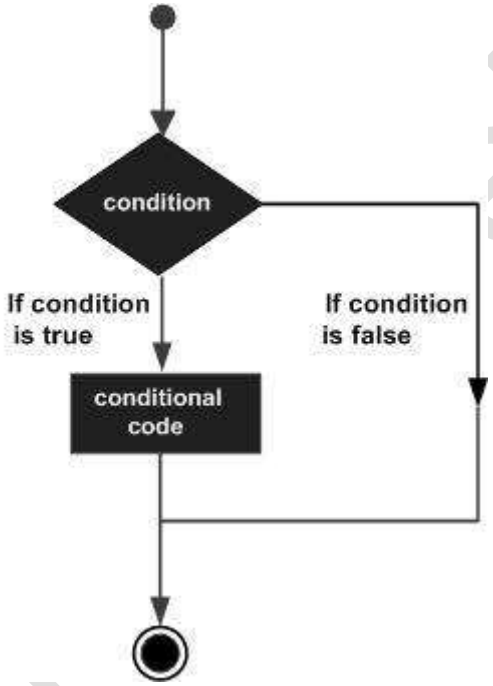
Dart Programming - Decision Making

Decision-making is the anticipation of conditions occurring during the execution of a program and specified actions taken according to the conditions.

Decision structures evaluate multiple expressions, which produce TRUE or FALSE as the outcome. You need to determine which action to take and which statements to execute if the outcome is TRUE or FALSE otherwise.

Following is the general form of a typical decision making structure found in most of the programming languages-



Dart programming language assumes any **non-zero** and **non-null** values as TRUE, and any **zero** or **null values** as FALSE value.

Dart programming language provides the following types of decision-making statements.

Statement	Description
if statements	An if statement consists of a Boolean expression followed by one or more statements.
if...else statements	An if statement can be followed by an optional else statement, which executes when the boolean expression is FALSE.
else if ladder	It is useful to test multiple conditions

switch..case statement	The switch statement evaluates an expression, matches the expression's value to a case clause and executes the statements associated with that case.

Let us go through each decision-making statement quickly.

IF Statement

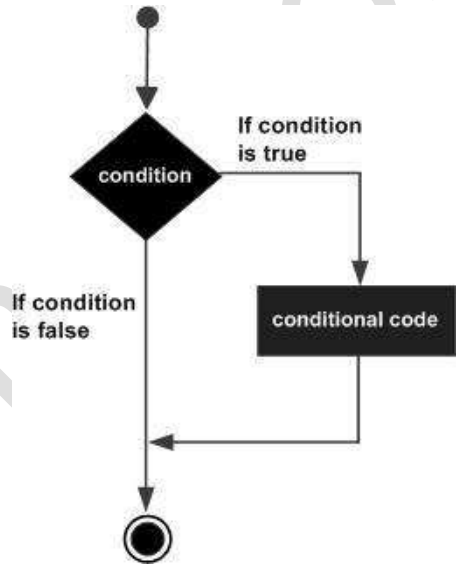
The IF statement is similar to that of other languages. The **if** statement contains a logical expression using which the data is compared and a decision is made based on the result of the comparison.

Syntax

```
if (expression){
  statement(s);
}
```

If the boolean expression evaluates to TRUE, then the block of statement(s) inside the if statement is executed. If boolean expression evaluates to FALSE, then the first set of code after the end of block is executed.

Flow Diagram



Example1

```
void main() {
   var  num=10;
   if (num>0) {
      print("$num number is positive");
   }
}
```

When the above code is executed, it produces the following result –

```
10 is positive number
```

Example2

```
void main() {
   var  num=0;
   if (num) {
      print("We got a true expression
value");
   }
   print(num);
   print("Good bye!");
}
```

When the above code is executed, it produces the following result –

```
0

Good bye!
```

IF...ELSE Statements

An **else** statement can be combined with an **if** statement. An **else** statement contains a block of code that executes if the conditional expression in the if statement resolves to 0 or a FALSE value.

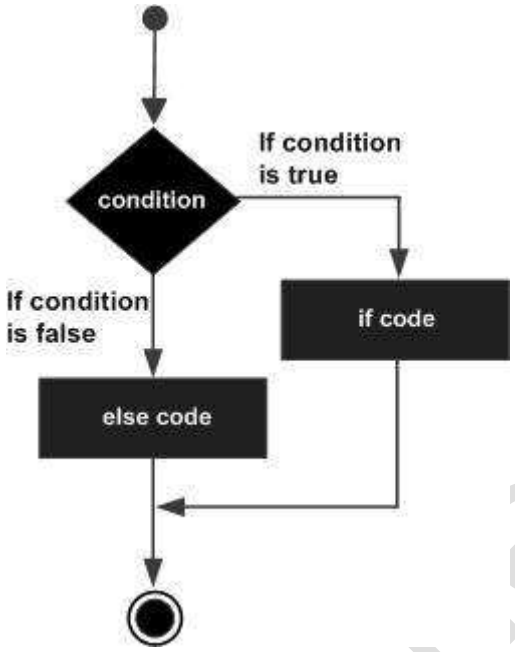
The else statement is an optional statement and there could be at the most only one **else** statement following **if**.

Syntax

The syntax of the **if...else** statement is-

```
if (expression){
   statement(s);
}
else{
    statement(s);
}
```

Flow Diagram



Example

```
void main() {
 var a;
 print("Enter number:");
 a=int.parse(stdin.readLineSync()!);
if(a>0){
    print ("$a is positive");
  }else{
    print ("$a is not positive");
  }
}
```

In the above example, If a is greater than zero then it print a is positive otherwise it print a is not positive. When the above code is executed, it produces the following result-

```
First Run:
Enter number:15
15 is positive
Second Run:
Enter number:-25
```

```
-25 is not positive
```

IF...ELSE IF...ELSE Statement

The **else if** statement allows you to check multiple expressions for TRUE and execute a block of code as soon as one of the conditions evaluates to TRUE.

Similar to the **else**, the **else if** statement is optional. However, unlike **else**, for which there can be at the most one statement, there can be an arbitrary number of **else if** statements following an **if**.

Syntax

```
if(expression1){
   statement(s);
}
else if(expression2){
   statement(s);
}
else{
   statement(s);
}
```

Example

```
void main() {
 var x,y,z;
 print("Enter first number:");
 x=int.parse(stdin.readLineSync()!);
 print("Enter second number:");
 y=int.parse(stdin.readLineSync()!);
 print("Enter third number:");
 z=int.parse(stdin.readLineSync()!);
 if (x>y && x>z){
    print ("$x is maximum number");
  }
  else if(y>z){
    print ("$y is maximum number");
  }
  else{
    print ("$z is maximum number");
  }
}
```

When the above code is executed, it produces the following result-

```
First Run:
Enter first number: 15
Enter second number: 25
```

```
Enter third number: 20
25 is maximum number

Second Run:
Enter first number: 45
Enter second number: 35
Enter third number: 28
45 is maximum number


Third Run:
Enter first number: 45
Enter second number: 35
Enter third number: 68
68 is maximum number
```

Switch Case Statements

Dart Switch case statement is used to avoid the long chain of the if-else statement. It is the simplified form of nested if-else statement. The switch statement evaluates an expression, matches the expression's value to a case clause and executes the statements associated with that case.

Syntax

The syntax of the switch case may be-

```
switch(expression) {
   case constant_expr1: {
      statement(s);
   }
   break;
   case constant_expr2: {
      statement(s);
   }
   break;
   default: {
      statement(s);
   }
}
```

Example

```
void main() {
            int n = 3;
```

```
            switch (n) {
                case 1:
                    print("Value is 1");
                    break;
                case 2:
                    print("Value is 2");
                    break;
                case 3:
                    print("Value is 3");
                    break;
                case 4:
                    print("Value is 4");
                    break;
                default:
                    print("Out of range");
                    break;
            }
        }
```

When the above code is executed, it produces the following result-

```
Value is 3
```

Explanation

In the above program, we have initialized the variable n with value 3. We constructed the switch case with the expression, which is used to compare the each case with the variable n. Since the value is 3 then it will execute the case-label 3. If found successfully case-label 3, and printed the result on the screen.